C868 Software Capstone Project Summary

Task 2 – Section A

Capstone Proposal Project Name: Ashlar CRM

Student Name: Daniel Berry

Table of Contents

Business Problem

The Customer

South Central Tennessee Tourism Association, or SCTTA, is one of nine regional non-profit tourism organization created by the State of Tennessee to assist the state with tourism development and marketing. SCTTA is responsible for a thirteen-county region that includes over twenty different cities. The Association's flagship marketing product is ExperienceTN.guide, is a tourism industry driven business listing and experience site. The association is partially funded by an endowment grant from the State of Tennessee, with additional funding coming from association memberships, the sale of marketing products, sponsorships on collateral distributed to welcome centers across the state, and through consulting services.

Tourism is the state's second largest industry behind agriculture. Being a member-driven regional association, establishing and maintaining relationships with tourism industry related businesses and organizations is critical to the overall success of the organization. Industry members are the backbone of the SCTTA and the Experience Tennessee Brand.

SCTTA is dedicated to their organizational mission of impacting local tourism-related revenue by driving traffic to hospitality industry businesses.

Business Case

As with most organizations, tracking the different stages of the sales cycle, tracking, and updating customer information, and providing a consistent way to track progress has been a challenge for the organization. The Association has used several different customer relationship management (CRM) applications such as Monday.com and Hubspot.com without finding one that both meets their needs and fits within their budget.

An easy-to-use, turn-key, web-based CRM that is budget-friendly and can grow and evolve as the association's needs change will benefit SCTTA by allowing them better maintain relationships and track the impact of their services.

Fulfillment

The CRM will be built to solve the issues mentioned in the business case section of this document. This will be a responsive web-based application built using PHP for the backend portion and JavaScript for interactivity on the front. It will be built on the Laravel 9 framework which is the latest long-term supported version. Using Laravel provides the application with database management, unit, and feature testing, and will power the API. VueJS 3 will be used as the JavaScript framework because of its wide-spread adoption and easy integration with the Laravel framework.

The application will connect to a remote MySQL database that will be hosted in the Amazon Web Services Cloud and will be driven by an API that will be integrated into the system. The API will allow the data to be used not only by this application, but also allow SCTTA to integrate the data into other applications and systems as the need arises.

The system will allow creating, updating, deleting, and the association of customer information including companies, contacts, deals, notes, and tasks. Reports will be available using searchable tables and baked-in standardized reporting functionality.

Existing Gaps

As previously mentioned, SCTTA has used several different CRMs in the past. The number one issue has been the monthly subscription costs to use the applications. As a non-profit organization, it is difficult to find the funding for internal-use applications. Switching from CRM to CRM has created an issue with data now being scattered across several platforms, so being able to enter this data into a centralized location is extremely important.

Because there is currently no centralized system individual employees and contractors are recording their information in disparate systems, there is no way for the association to get a clear big picture view. Also, as employees and contractors leave, the data they gathered is often lost with them.

SDLC Methodology

We will be utilizing the Agile SDLC Methodology in this project because of the need to engage the customer during the development cycle for feedback due of the highly customized nature of the product and the customer's immediate need. Utilizing the agile methodology will allow us to break the application into smaller feature-based iterations that can be tested by developers and the customer in intervals versus completing that step at the end of the project. Because this is a web-based application, a staging server will be used for testing purposes with new features being added at the end of the sprint.

Utilizing the agile methodology will also allow us to deliver the application feature by feature so that verifying and validating feature requirements can happen along the way versus all at the end. As features are completed, the end-user can be involved in prioritizing the next feature set to be released.

Overall, the agile methodology allows us to deliver the application incrementally instead of delivering the entire product at once in the end and allows us to shift as needed. This method will include the following phases: concept, inception, iteration, release, and maintenance.

Conception

During this stage, an overall project assessment will be conducted with the stakeholders to determine the time and resources required for the development process. The product owner will also assess the risks and prioritize functionality based on their business value.

Deliverable: Documented minimum requirements and approval from the client to begin work.

Inception

Now that we have an outline of the concept, the product owner will meet with the team to introduce the requirements. The development team will build the product

architecture, create a mockup of the user interface, and flesh out the requirements on a diagram.

Deliverable: UI mockup, entity diagram, and initial development environment.

Iteration

Now that we have a prioritized list of features based on scope and requirements, and have UI mockups and the development environment setup, the team will begin developing the product itself. The product will be delivered in separate stages, or sprints, with each sprint improving the previously iterated version of the product. Each sprint will begin by defining the goals for the sprint and will end with both white and black box testing of the features being delivered. Again, because we have chosen the agile methodology, the stakeholders will be able to review the features incrementally and make changes to the requirements or functionality as the project progresses instead of waiting until the end where changes are more costly.

Deliverable: A software product the meets the requirements, diagrams, and business goals.

Release

With all sprints and iterations being complete, the application will undergo quality assurance testing to ensure the software is both fully functional and meets the requirements specified during the conception and inception phases. Bugs and defects will be assigned to the development team to be addressed in a sprint cycle. User training and documentation will also take place during this phase.

Once all testing and training has been completed, the product will be fully released into production.

Deliverable: Completed test plan, user documentation and end-user training.

Maintenance

With the software being fully deployed and user training completed, the development team will begin the maintenance phase. During this phase, the software team will provide ongoing maintenance and support. Any bugs or defects will be reported and resolved, and new features and product upgrades will occur as the need arises.

Deliverable: Continued support and maintenance, bug fixes, and new features as needed.

Deliverables

Different deliverables are included in each of the phases above. This section is provided to give a clearer picture of each. These have been broken down into two different sections. Based on our agile approach, we have broken the deliverables into project and product deliverables.

Project Deliverables

These consist of items that are part of the Project Manager's realm of responsibilities.

- Minimum Requirements Document
    - This document will contain a list of client-approved requirements. Because we are using the agile methodology, this list will contain the minimum requirements that set the overall scope of the project.
- Client Approval Form
    - This document contains the project's scope, estimated timeline, and costs. The client will be required to review and approve this document before the project begins.
- Wireframe and UI Mockup
    - The frontend developer will provide a low-fidelity mockup that is used to validate the application's flow. Wireframes describing the basic functionality will be created for each screen. The client will approve the wireframe and mockup before proceeding to the next phase.

- Entity Relationship Diagram
    - This diagram shows the relationships of the different entities stored in our database. The diagram will show each entity's attributes and show the relationships that exist between them.
- Development Environment
    - We will be using a a docker container for our development and production environment. This ensures that the developers, quality assurance team, and support groups are all working off an environment that matches our production environment.
- Testing Plan
    - The testing plan describes the testing methodology and procedures that will be used to validate that the application is free of bugs and meets the desired business goals.
- End User Documentation
    - End user documentation will be provided that will be used as a reference guide and instruction manual for navigating and using the system.

Product Deliverables

Product Deliverables represents what is produced to deliver to the customer.

- Fully functional web-based customer relationship management application
    - The final application, built to the requirements, will be a scalable multi-tenant cloud-based application deployed in an always available environment. The user-friendly interface will allow users to access customer and contact information, record new deals and opportunities, and add notes and tasks to each of the primary modules.
- API to access the data from other applications and systems
    - A secure API will be both consumed by the application itself and made available to the customer so that the data in the application can be accessed by existing or new applications and in-house systems.
- Secure access
    - Team admins will have the ability to add new users to the system with various roles and access permissions. In addition to standard username and password authentication, users will have the option to turn on two-factor authentication for added security.

Implementation

The implementation of this application is a simple process. Because the application is a new cloud-based application, there is no need for individual deployment, no downtime or outages, and the deployment will be staged prior to onboarding the customer.

The docker containers used to develop the application will also be used when deploying to production. This ensures that the same software and environment that is validated by our quality assurance team, is the same environment being deployed.

Roll out to the production servers will take place once the quality assurance team has validated that the application is free of defects and bugs and meets the requirements and business needs. Once the production server is deployed, the QA team will run smoke testes to validate the rollout. If the rollout passes the smoke test, we will coordinate with the client stakeholder to verify the install and setup the first team and begin inviting users to the application. At this point the production rollout is complete and we will begin the maintenance phase.

Validation and Verification

Due to our agile process, testing and verification of features will take place throughout the entire software development lifecycle. At the end of each sprint cycle, the developers will walk through the feature with the stakeholders to ensure the business need has been met, and the QA team will run through the feature's testing plan to ensure there are no bugs or defects.

At the end of the Iteration stage in the product lifecycle, the team will enter the release stage. During the release stage the team will walk through the entire application with the stakeholders to again verify the business goals and expectations have been met. The QA team will go through the entire test plan and application to ensure all requirements have been met and catch any remaining bugs or defects.

Once all testing is complete, the customer will sign off and accept the application validating that the application looks and works as expected.

Environments and Costs

Programming Environment

The application will be built using the following environment:

- Ubuntu 21.10 or later
- PHP 8.1
- MySQL 8.0 or later
- Laravel PHP Framework, version 9.11.*
- InertiaJS 0.11.0
- Docker for the development environment
- Redis 6.2.2

Environment Costs

Because this application is web-based, it will be deployed to cloud hosting provided by Digital Ocean which operates on the AWS platform. Initial environment costs will be minimal, but because this is a cloud-based architecture, the application can scale as needed. Below is a table explaining the initial monthly costs:

SERVICE	PROVIDER	SPECS	MONTHLY COST
Webserver	Digital Ocean – US East	2 GB RAM, 1, Intel CPU, 50GB NVMs SSD HDD, 2 GB Transfer	$12.00
MySQL Database	Digital Ocean – US East	1 vCPU, 1 GB RAM, 10 GB SSD, Connection limit 75.	$15.00
Redis Database (caching)	Digital Ocean – US East	1 vCPU, 1 GB RAM, 10 GB SSD,	$15.00
		TOTAL MONTHLY	**$42.00**

Human Resource Requirements

We will assemble a team consisting of a team lead, a frontend designed, backend developer, and a quality assurance engineer to complete the project. The backend

developer consumes the largest human resource expense followed by team lead. Below is a table detailing each of the roles, with their hourly rate, an estimate of the time involved, and a total for each.

Role	Rate	Estimated Time	Total
Team Lead	$75/hour	70 hours	$5,250
Frontend Designer	$50/hour	30 hours	$1,500
Backend Developer	$75/hour	120 hours	$9,000
QA Engineer	$50/hour	60 hours	$3,000
		TOTAL	**$18,750**

Project Timeline

Below is a table describing the project timeline. Because we are using the agile methodology, several of the tasks will be run asynchronously.

Phase	Milestone/Task	Deliverable	Description	Dates
Concept	Task 1	Minimum Requirements	Meeting with the client to conceptualize the application and gather the minimum set of requirements	1/2/22 – 1/10/22
Concept	Task 2 / Feature Prioritization	Prioritized list of features, and approval from the client to begin work.	Client will provide a list of the features based on their business value	1/11/22 – 1/15/22

Concept	Task 3 / Client Approval	Approval from client to being work	Client will sign approving the timeline and costs	1/16/22 – 1/18/22
Inception	Task 4 / Wireframe and UI mockup	Wireframe and UI mockup	Design and validate the application's user interface	1/19/22 – 1/27/22
Inception	Task 5 / Entity Diagram	Diagram depicting the database entities	This diagram will be used to build the database schema and defined relationships	1/19/22 – 1/27/22
Inception	Task 6 / Environment setup	Development environment	The team will deliver a docker-based development environment that mimics the production environment.	1/19/22 – 1/27/22
Iteration	Task 7 / Application development and initial delivery	Fully web-based CRM application	The team will deliver a software application that meets the requirements, diagrams, and business goals	1/28/22 – 3/1/22
Release	Task 8 / QA testing	QA performs tests and gives final approval to release the application	The QA engineer will perform all required tests and verify the functionality in each area of the application	3/2/22 – 3/15/22

Release	Task 9 / Documentation	User documentation	While QA is testing the application, the team will develop the end-user documentation.	3/2/22 – 3/15/22
Release	Task 10 / Training	End-User Training	The end-users will receive the documentation and hand-on training on how to use the application.	3/16/22 – 3/19/22
Release	Task 10 / Deployment	The application will be deployed to the cloud-based production servers.	The web, database, and caching servers will be setup to the same specs as the Docker development environment and the application will be deployed.	3/20/22 – 3/21/22
Maintenance	Task 11 / Maintenance Plan	On-going support and maintenance contract	Terms will be discussed with the client for on-going customer support and maintenance.	3/22/22 +